

## Contents

<b>1 Routine/Function Prologues</b>	<b>2</b>
1.0.1 retagrlw.F90 (Source File: retagrlw.F90) . . . . .	2

## 1 Routine/Function Prologues

### 1.0.1 retagrlw.F90 (Source File: retagrlw.F90)

Opens, reads, interpolates and overlays LW radiation forcing.

TIME1 = most recent past data

TIME2 = most recent future data

1. Load AGRMET 3 level cloud amount (converted from RTNEPH)
2. Convert 2 m forcing specific humidity to vapor pressure
3. Use AGRMET subroutine longwv() to calculate LW DOWN. Arguments are:  
cloud amount [%], 2 m temperature [K], and 2 m vapor pressure [Pa]

#### REVISION HISTORY:

```

28 Oct 1999: Brian Cosgrove; Initial code
11 Apr 2000: Brian Cosgrove; changed code to use Forcing Mask (With inland
              water filled in). Deleted unused variables.
20 Jun 2000: Brian Cosgrove; changed code so that it uses LDAS%UDEF and
              not a hard-wired undefined value of -999.9 and -999.0
25 Jun 2001: Urszula Jambor; Modified for AGRMET data use in GLDAS.
25 Oct 2001: Jesse Meng; Modified for AGRMET LW scheme implementation

```

#### INTERFACE:

```
subroutine retagrlw ( order, yr, mo, da, hr, ferror, flag )
```

#### USES:

```

use lisdrv_module, only : lis, grid
use obsradforcing_module, only : oblwdatal, oblwdat2
use agrmetdomain_module, only : agrmetdrv
implicit none

```

#### ARGUMENTS:

```

integer :: yr, mo, da, hr
integer :: order                      !retrieve data for time1 or time2
integer :: ferror                     !0=no radiation data found
                                         !1=found observational data

```

#### CONTENTS:

```

!-----
! If using AGRMET data, open, read in and interpolate AGRMET files
! to appropriate GLDAS resolution;
! If error reading file, or data completely missing, ferror=0;
!-----
ferror = 0
do c = 1, lis%d%ngrid

```

```

if (order == 1) then
    oblwdatal(c) = lis%d%udef
else if (order == 2) then
    oblwdatal2(c) = lis%d%udef
end if
end do

if (flag == 1) then
!---
! 1. Generate AGRMET cloud filenames
! Load 3 layers cldamt data
!---

write(cyr, '(I4.4)') yr
write(cmo, '(I2.2)') mo
write(cda, '(I2.2)') da
write(chr, '(I2.2)') hr

nameSH = trim(agrmtdrv%agrmtdir)//'/CloudAGR//cyr//cmo// &
        '/cldamt_//cyr//cmo//cda//chr
print*, 'Reading AGRMET file : ',nameSH
open(11, file=nameSH, status='old',access='direct',&
      recl=3456000, iostat=openerrN)
read(11, rec=1,iostat=readerrN) pdata1H
read(11, rec=2,iostat=readerrN) pdata1M
read(11, rec=3,iostat=readerrN) pdata1L
close(11)
if ((openerrN+readerrN) > 0) then
    ferror = 0
    print*, 'AGRMET file problem: ',nameSH
else
    ferror = 1
endif
call interp_agrmel_lw( pdata1H,cldamtH, ferror )
call interp_agrmel_lw( pdata1M,cldamtM, ferror )
call interp_agrmel_lw(pdata1L,cldamtL, ferror )
if ( ferror .EQ. 0 ) return
!---
! If AGRMET cloud data is not undefined, set ferror=1
!---

fvalid = 0
do c=1, lis%d%ngrid
    if (cldamtH(c) >= 0.0) fvalid = 1
    if (cldamtM(c) >= 0.0) fvalid = 1
    if (cldamtL(c) >= 0.0) fvalid = 1
end do
if ( fvalid .EQ. 0 ) return
do c=1,lis%d%ngrid

```

```

      rldown = 0.
      tair = grid(c)%forcing(1)
!-----
! 2. CONVERT 2 METER SPECIFIC HUMIDITY TO VAPOR PRESSURE
!-----
      vaporP = grid(c)%forcing(2) * grid(c)%forcing(7) / &
                 ( 0.622 + grid(c)%forcing(2) * (1-0.622) )
!-----
! If tair, vaporP, and ALL 3 AGRMET LAYERS cldamt are defined,
! calculate rldown; transfer to proper output array
!-----
      fvalid = 1
      if (tair.LT. 0.0) fvalid = 0
      if (vaporP.LT. 0.0) fvalid = 0
          if (cldamtH(c) .LT. 0.0) fvalid = 0
          if (cldamtM(c) .LT. 0.0) fvalid = 0
          if (cldamtL(c) .LT. 0.0) fvalid = 0

      if (fvalid == 1) then
          cldamt1d(1) = cldamtL(c)
          cldamt1d(2) = cldamtM(c)
          cldamt1d(3) = cldamtH(c)
!-----
! 3. Calculate lw down
!-----
      call agrlwdn( tair, vaporP, cldamt1d, rldown )
      if (order==1) then
          OBLWDATA1(c) = rldown
      else if (order==1) then
          OBLWDATA2(c) = rldown
      end if
      else
          if (order==1) then
              OBLWDATA1(c) = lis%d%udef
          else if (order==1) then
              OBLWDATA2(c) = lis%d%udef
          end if
      end if
      end do
  end if
  return

```